University of Houston

COSC 6377
Computer Networking

Dr. Omprakash Gnawali

Homework 1

Ashutosh Kumar
1854393

Question 1

Suppose you are hired as a network manager at the university. The university is planning on enforcing a policy to prevent all users (faculty, students, staff, guests) from accessing the website `tiktok.com` while using the campus network. Propose two solutions to implement this policy in the campus network. Provide sufficient explanation on how each solution prevents users from accessing the website.

Solution. Two ways to block the access of a given website from the users of a network are as follows.

1. **Blocking the DNS request**
   When a user tries to access a website, their device requests a DNS for the IP address for the server containing the website by giving it the domain name.

   The packets being sent to the DNS to request the website to be blocked (`tiktok.com` in this case) can simply be dropped by the router that connects the university network to the public internet. This way, the DNS never receives the request, so the user never gets the IP address for the server, hence cannot access the website.

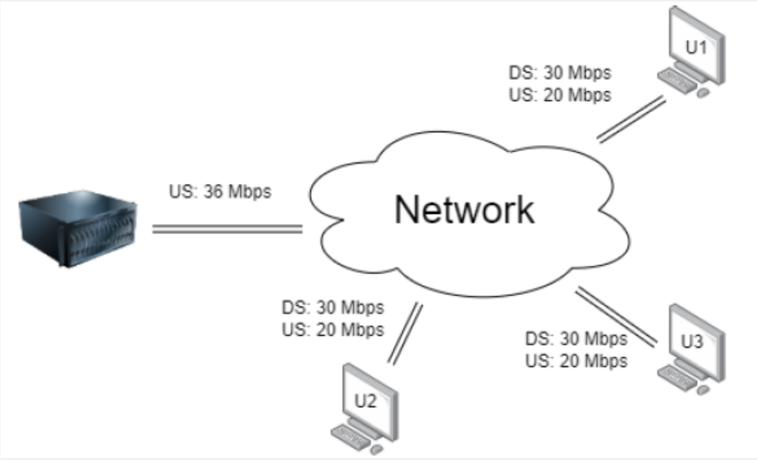
   Additionally, the packets never leave the university network to enter the public internet, so we do not waste any bandwidth on the public internet.

2. **Configuring Firewall Rules**
   Another way to block a website it to set Firewall rules in the router connecting the internal network and the public internet. We can have the router drop the packets coming from the servers that serve the website to be blocked. This can be done by adding the IP addresses of these servers to the firewall rules.

   However, since the IP addresses can change and a single website can be (and usually is) served by multiple servers, all the IP addresses must be added to the firewall rules, and they must be regularly updated.

Question 2

Consider the network below. The server contains a file with the size of 30 Gigabits. Users U1, U2, and U3 are trying to download this file from the server. The Upload Speed (US) and the Download Speed (DS) of each link is also shown in this figure. U1 starts downloading at time t=0s, U2 starts at t=10s, and U3 starts at t=20s.



a) If the users and the server use the client/server architecture, what will be download time for each user. You can assume that the server divides the bandwidth equally among all the users that are connected to the server.

b) To reduce download time, users agree to implement a peer-to-peer protocol where each user shares the segments of the file it has with other users and users only access the server for the remaining segments that are still unavailable. If U1, U2, U3 start downloading at 0s, 10s, 20s using this peer-to-peer protocol, how much improvement will they get compared to the client/server approach.

Solution. The following image 1 visualises the timeline described in the question.

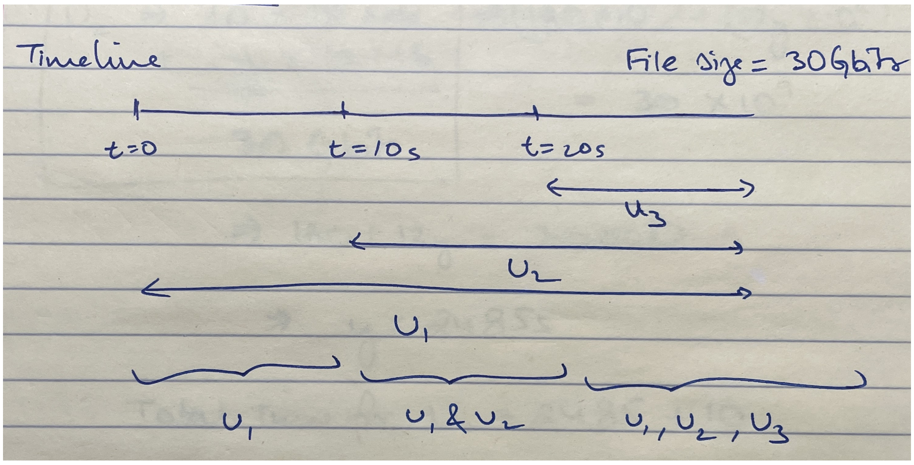


Figure 1: Timeline for Question 2

a) **Client-Server Architecture**
First, I will describe in words how much bandwidth each of the user gets and for how long, then I will describe the same mathematically.

- For the first ten seconds, `U1` gets the bandwidth of 30 Mbps (limited by the download speed of `U1`).
- For the next ten seconds, the bandwidth is equally divided between `U1` and `U2`. That is, both `U1` and `U2` download at the rate of 18 Mbps.
- Then, for some amount of time (different for all the users), each user gets to download at the rate of 12 Mbps.

Now, let us look at the time taken by each user to download the 10 Gigabit file.

- For `U1`, the total time taken will be $(10 + 10 + x)$ seconds. Let's find the value of x first.

$$
\begin{aligned}
(10 \text{ s} \times 30 \text{ Mbps}) + (10 \text{ s} \times 18 \text{ Mbps}) + (x \text{ s} \times 12 \text{ Mbps}) &= 30 \text{ Gbits} \\
(300 \times 10^6) + (180 \times 10^6) + (12x \times 10^6) &= 30 \times 10^9 \\
12x &= 30000 - 480 \\
x &= 2460 \text{ s}
\end{aligned}
$$

Therefore, the time taken by `U1` is $\boxed{2480s}$.

- For `U2`, the total time taken will be $(10 + y)$ seconds. Let's find the value of x first.

$$\begin{aligned}(10 \text{ s} \times 18 \text{ Mbps}) + (y \text{ s} \times 12 \text{ Mbps}) &= 30 \text{ Gbits} \\ (180 \times 10^6) + (12y \times 10^6) &= 30 \times 10^9 \\ 12y &= 30000 - 180 \\ y &= 2485 \text{ s}\end{aligned}$$

  Therefore, the time taken by `U2` is $\boxed{2495s}$.

- For `U3`, the total time taken will be z seconds. Let's find the value of z.

$$\begin{aligned}z \text{ s} \times 12 \text{ Mbps} &= 30 \text{ Gbits} \\ 12z \times 10^6 &= 30 \times 10^9 \\ 12y &= 30000 \\ z &= 2500 \text{ s}\end{aligned}$$

  Therefore, the time taken by `U3` is $\boxed{2500s}$.

b) **Peer-to-Peer Approach**

Since the peer sharing technique is not mentioned in the question, we will assume that all the users share files to each other. How they decide what part of the file which one of them downloads from the server, we will let the users decide (not in the scope of this question). The bandwidth will be divided equally amongst the users, that is, the upload bandwidth from one user to multiple users will be equally divided.

First, we will visualise how the file transfer happens, and then we will look at the mathematics.

From time $t = 0 \rightarrow 10$s, only `U1` exists, so `U1` gets the entire bandwidth of 30 Mbps (limited by the download speed of `U1`). This can be seen in the image 2.

Figure 2: Bandwidth division from time $t = 0 \rightarrow 10$s

From time $t = 10 \rightarrow 20$s, users `U1` and `U2` are downloading the file, so each of them gets a bandwidth of 18 Mbps from the server. From each other, the users get a bandwidth of 12 Mbps (this follows the upload bandwidth constraints of the users, as each of them is using only 12 Mbps out of 20 Mbps). So in total, each of the users still gets 30 Mbps bandwidth. This can be seen in the image 3.

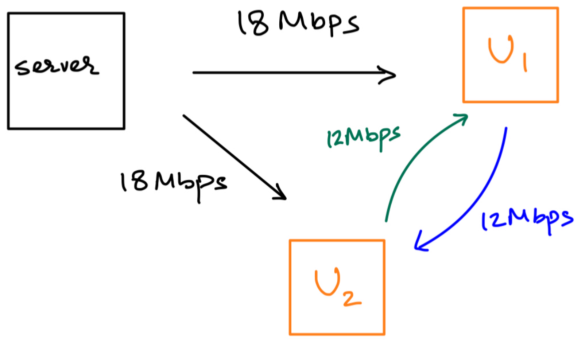


Figure 3: Bandwidth division from time $t = 10 \rightarrow 20$s

From time $t = 30$s onward, users `U1`, `U2`, and `U3` are downloading the file, so each of them gets a bandwidth of 12 Mbps from the server. From each other, the users get a bandwidth of 9 Mbps (this follows the upload bandwidth constraints of the users, as each of them is using 18 Mbps out of 20 Mbps). So in total, each of the users still gets 30 Mbps bandwidth. This can be seen in the image 4.

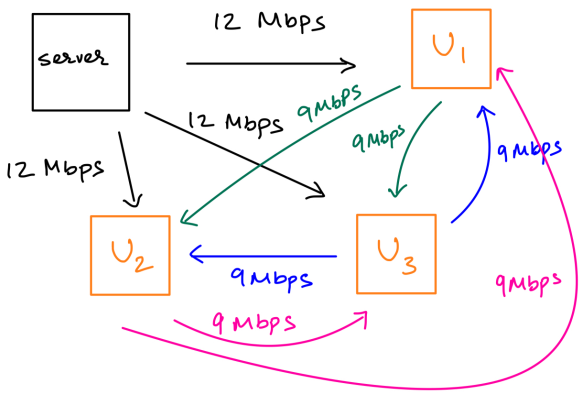


Figure 4: Bandwidth division from time $t = 30$s onwards

Now, we will calculate the time taken by each user to download the file.

- For `U1`, it will take $(10 + 10 + x)$ seconds to download the file. Let's find the value of x.

$$(10 \text{ s} \times 30 \text{ Mbps}) + (10 \text{ s} \times 30 \text{ Mbps}) + (x \text{ s} \times 30 \text{ Mbps}) = 30 \times 10^9$$
$$600 + 30x = 30000$$
$$x = 980 \text{ s}$$

  Therefore, the time taken by `U1` is $\boxed{1000s}$.

- For `U2`, it will take $(10 + y)$ seconds to download the file. Let's find the value of y.

$$(10 \text{ s} \times 30 \text{ Mbps}) + (y \text{ s} \times 30 \text{ Mbps}) = 30 \times 10^9$$
$$300 + 30y = 30000$$
$$y = 990 \text{ s}$$

  Therefore, the time taken by `U2` is $\boxed{1000s}$.

- For `U3`, it will take z seconds to download the file. Let's find the value of z.

$$\begin{aligned} z \text{ s} \times 30 \text{ Mbps} &= 30 \times 10^9 \\ 30z &= 30000 \\ z &= 1000 \text{ s} \end{aligned}$$

Therefore, the time taken by `U3` is $\boxed{1000s}$.

Compared to the Client-Server Architecture, we see significant improvements in the download time for each of the three users. User `U1` saves 1480 seconds. User `U2` saves 1495 seconds. And user `U3` saves 1500 seconds.

Question 3

To increase security, the University of Houston is planning on restricting the access of the guest users. In this policy, guest users are only allowed to use the campus network to access the Internet and some of the public University websites. They are also not allowed to access the content of the packets sent from faculties or students. The UH IT department asks you to propose a solution to implement this policy. In the current network infrastructure, all network users and servers are connected to the same network and the university is reluctant to change the current physical topology. However, you are free to make any software change. What is going to be your proposed solution for this problem?

Solution. First, we have to distinguish guest users from students and faculty. This can be done by having the students and faculty log in with their university email to access the internet. Guest users can log in using their email/phone number.

Since the guest devices will be assigned an IP address when they join the networking, we can use that to restrict their access. The router at the edge of the network can drop the packets from these devices if they try to access restricted websites (using firewall rules or by dropping DNS request packets by these guest devices).

Similarly, any packet from students/faculty trying to get to these guest devices can be dropped by the internal routers/switches (using Access Control Lists (ACLs) and MAC addresses).

Question 4

Node A wants to send a 10 GB file to node B. The round trip time between the nodes is 200 milliseconds and the data rate of the link between the nodes is 100 Mbps. Suppose that the maximum frame size is 1020 bytes and packet header size is 20 bytes. The transmission time (not the propagation time) of the ACK frame is negligible.

a) Without any `ack` messages to ensure reliability, how long does it take for A to send the file to B.

b) If both nodes decide to use stop-and-wait protocol with window size of one, what will be the new file transfer time?

c) If nodes decide to increase window size to 100, what will be the new transmission time?

d) What is the best window size that minimizes the file transfer time?

Solution. A 10 GB file has 10×10^9 bytes.

a) Time taken is defined as the file size divided by the link rate.

$$\text{time} = \frac{8 \times 10 \times 10^9}{100 \times 10^6}$$
$$\boxed{\text{time} = 800 \text{ s}}$$

b) If the nodes decide to use the stop-and-wait protocol with the window size of one, it will take the sender 200ms to send a packet (1000B of actual data + a 20B header) and receive the `ack`. The effective data rate here is 5000 B/s.

$$\begin{aligned}\text{time} &= \frac{10 \times 10^9}{5000} \\ &= \frac{10000 \times 10^6}{5000}\end{aligned}$$
$$\boxed{\text{time} = 2 \times 10^6 \text{ s}}$$

c) In this case, the nodes decide to use the stop-and-wait protocol with the window size of 100. Assuming negligible time taken to put the packets onto the wire (negligible transmission time), it will take 200ms to send 100×1000B of actual data. Therefore, the effective data rate here is 5×10^5 B/s.

$$\text{time} = \frac{10 \times 10^9}{5 \times 10^5}$$
$$\boxed{\text{time} = 2 \times 10^4 \text{ s}}$$

d) From the perspective of the sender and considering that the network is quite reliable, the best window size would cover the *entire* file.

However, due to a variety of reasons, packets drop or do not reach the destination or timeout, so it would be prudent to have a window size that is big enough that waiting for the `ack` messages do not slow down the file transfer time and at the same time, it should be small enough that resending an entire window's worth of data does not hinder the throughput (considerably).

Question 5

We are developing an implementation of a TCP protocol, and we are using EWMA to estimate TCP RTT. However, in our code, there is a bug that reduces the sample RTT by 20%. The table below describes the RTTs measured during a TCP trace:

Step	Measured RTT
1	20 ms
2	30 ms
3	25 ms
4	22 ms

a) Considering the bug in the implementation, what is the estimated RTT at step 5.

b) What is the impact of this bug on the performance of TCP?

Solution. Due to the bug in the code, the new sample RTT are given in the following table.

Step	Measured RTT (in ms)	New RTT (in ms)
1	20	16
2	30	24
3	25	20
4	22	17.6

a) The formula we will use is

$$\text{Estimated RTT}(t) = (\alpha \times \text{Sample RTT}) + ((1-\alpha) \times \text{Estimated RTT}(t-1))$$

Let's calculate the estimated RTT for each step.

Step 1: $(0.3 \times 16) + (0.7 \times 0) = 4.8$ ms

Step 2: $(0.3 \times 24) + (0.7 \times 4.8) = 10.56$ ms

Step 3: $(0.3 \times 20) + (0.7 \times 10.56) = 13.392$ ms

Step 4: $(0.3 \times 17.6) + (0.7 \times 13.392) = 14.6544$ ms

Step 5: $(0.3 \times \text{Sample RTT for this step}) + (0.7 \times 14.6544)$ ms

Because of the error in the code, we are underestimating the above values. With the correct values, the estimates should be as follows.

Step 1: $(0.3 \times 20) + (0.7 \times 0) = 6$ ms

Step 2: $(0.3 \times 30) + (0.7 \times 6) = 13.2$ ms

Step 3: $(0.3 \times 25) + (0.7 \times 13.2) = 16.74$ ms

Step 4: $(0.3 \times 22) + (0.7 \times 16.74) = 18.318$ ms

Step 5: $(0.3 \times \text{Sample RTT for this step}) + (0.7 \times 18.318)$ ms

As we can see, the 'true' estimates are larger than the estimates that our code found.

b) Due to the bug, as we can see in part (a), we underestimate the RTT. Because of this, we will set our timeout to be less than what it should be. The consequence of this is that the retransmission timeout will often be triggered even if the packet is still underway. Retransmissions will only serve to congest the network.